263-2300-00: How To Write Fast Numerical Code
Assignment 6
Due Date: Thu May 12 17:00
http://www.inf.ethz.ch/personal/markusp/teaching/263-2300-ETH-spring11/course.html

Code template: The code templates you can find on the course website or directly here.

Submission instructions: For tasks 2 and 3 use the template Intrinsics.c and for task 1 and 4 use the file MVM/main.c. Submit the final code into your SVN directory into a corresponding subfolder of hw06.

You can assume that all arrays lengths are divisible by 4 and that they are 16-byte aligned. Do not change the signature of the functions provided.

1. *(20 pts)* **Getting icc to work**

   The Intel compiler can be downloaded for evaluation here: Intel Software Evaluation Center.

   To make sure your compile environment works properly, you must compile the file `MVM/main.c` that we provide. A Visual Studio solution for windows and a makefile for linux/mac is also provided. The `/FAs` flag under windows generates a .asm file with comments. Under linux/mac the equivalent flag is `-S` to generate a .s file. Lookup which kind of instructions the first warm up (marked with comments) is translated into and list them as comments in your submitted code in task 4. Submit the .s or .asm file to the svn as well.

2. *(30 pts)* **Intrinsics: Warmup** (use the template in Intrinsics.c code)

   (a) Using `_mm_add_ps`, implement a vectorized function with parameters `float *y`, `float *x1`, `float *x2` and `float *x3` that computes $y = x1 * x2 + x3$.

   (b) Using `_mm_set1_ps`, implement a vectorized function with parameters `float *y`, `float *x` and *a constant* `float a` that computes $y = a * x$.

   (c) Using `_mm_set_ps`, implement a vectorized function with parameters `float *y`, `float *x` that is equivalent to the following scalar code:

   ```
   for (i=0;i<N;i++)
      Y[i]=i*i*X[i];
   ```

3. *(25 pts)* **Intrinsics: Medium** (use the template in Intrinsics.c code)

   Implement a pointwise multiplication of two complex vectors $(c_0, \ldots, c_{n-1})$ of length n by a second complex vector of the same length. Each array of complex numbers is represented as an array of twice the length containing floats. This array contains alternating the real and imaginary parts:

   $$(\mathrm{Re}(c_0), \mathrm{Im}(c_0), \ldots, \mathrm{Re}(c_{n-1}), \mathrm{Im}(c_{n-1})).$$

   This format is called *interleaved* complex format.

   Look at the scalar version in Intrinsics.c and implement a vectorized version with the same interface.

4. *(25 pts)* **MVM** Implement a vectorized version of an $n \times n$ matrix vector multiplication using the template provided in MVM/main.c. The code should be self-explanatory.